Improved Cryptanalysis of the Reduced Grøstl Compression Function, ECHO Permutation and AES Block Cipher

Florian Mendel[1], Thomas Peyrin[2], Christian Rechberger[1], *Martin Schläffer*[1]

IAIK, TU Graz, Austria

Ingenico, France

Overview

Motivation

- NIST SHA-3 Competition
  - many design strategies
  - different AES based hash functions
- Improve the cryptanalysis of AES based designs
  - additional degrees of freedom
  - different attack strategies
- Improve the security of AES based designs
  - how far can we go?
  - how much do we need?

Collision Attacks on Compression Functions

- iterated hash function $h(M, IV)$
  - compression function f: $H_t = f(M_t, H_{t-1}),\ H_0 = IV$

(1) collision:
- fixed IV
- $f(M_t, IV) = f(M'_t, IV),\ M_t \neq M'_t$

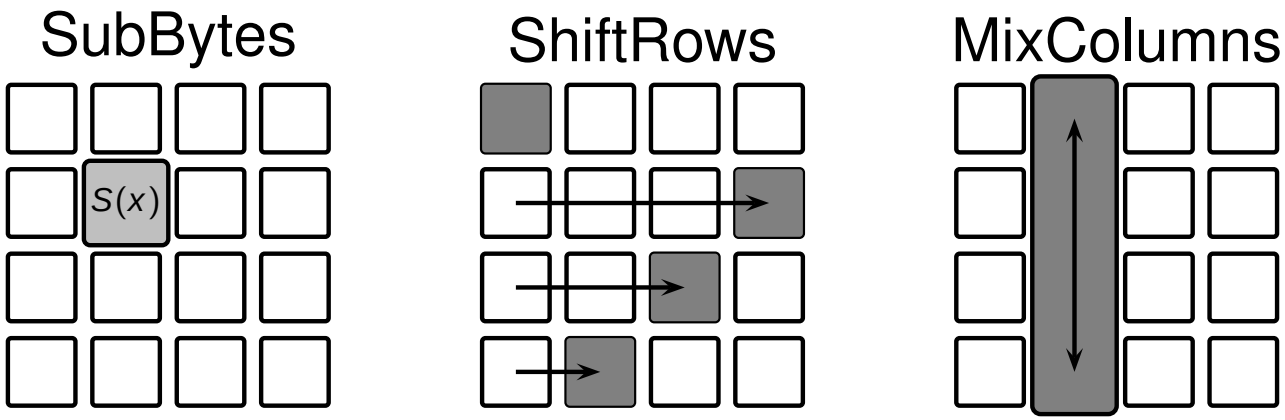
(2) semi-free-start collision:
- random chaining input
- $f(M_t, H_{t-1}) = f(M'_t, H_{t-1}),\ M_t \neq M'_t$

(3) free-start collision:
- random differences and values of chaining input
- $f(M_t, H_{t-1}) = f(M'_t, H'_{t-1}),\ M_t \neq M'_t, H_{t-1} \neq H'_{t-1}$

$\Rightarrow$ increasing degrees of freedom

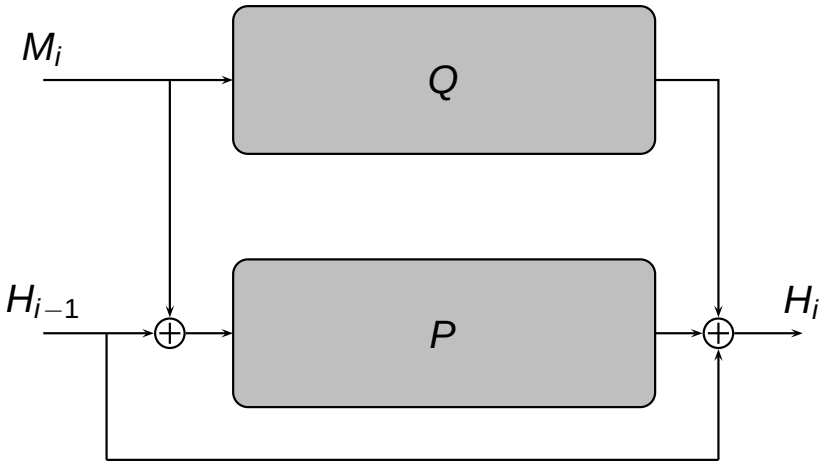
Overview

The Advanced Encryption Standard



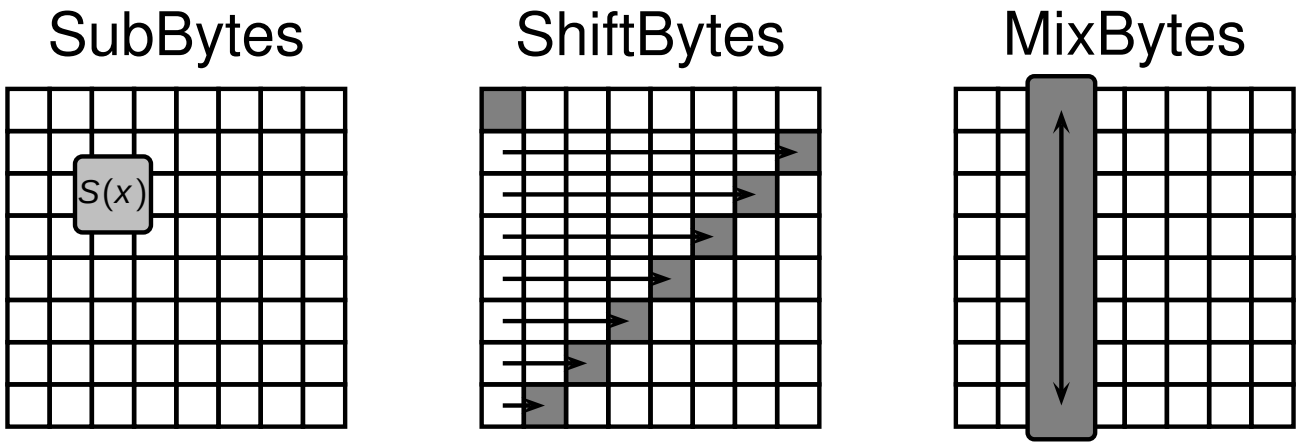
- state: 4×4 bytes
- AES round transformations:
  - $r_i = MC \circ SR \circ SB \circ AK$
  - $r_n = AK \circ SR \circ SB \circ AK$
- AES in known-key setting
  - state update behaves as a permutation

SHA-3 Candidate Grøstl



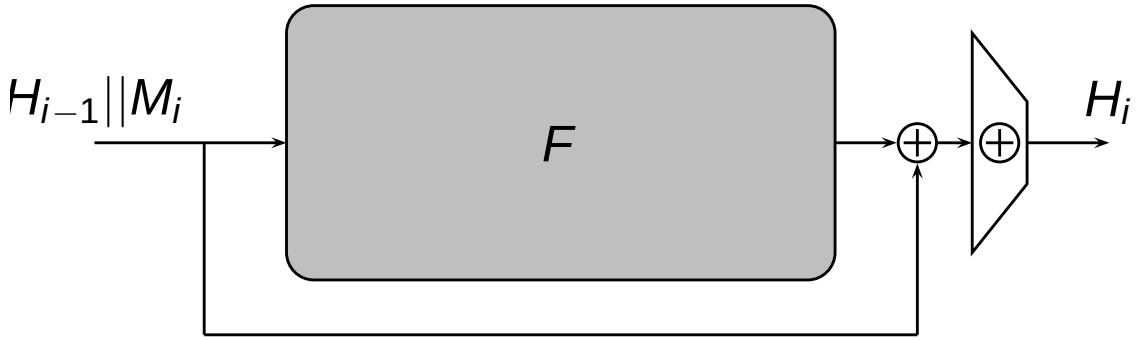
- Compression function of Grøstl
  - permutation based
  - AES based round transformations
  - no key-schedule inputs

The Grøstl-256 Permutations



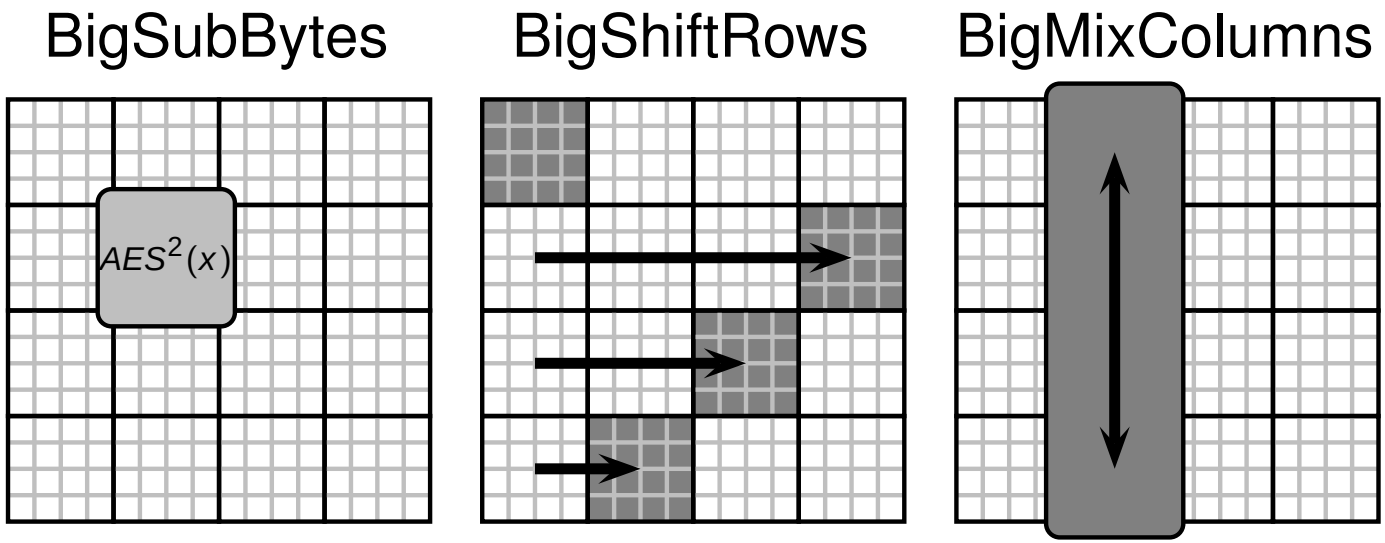
- Grøstl-256:
  - state: 8×8 bytes for P_{512} and Q_{512}
  - different round constants
  - 10 rounds each

SHA-3 Candidate ECHO



- Compression function of ECHO
  - permutation based
  - AES based round transformations
  - no key-schedule inputs

The Round Transformations of ECHO



- ECHO permutation:
  - 4×4 AES states
  - BigSubBytes:
    - 128-bit S-box (two AES rounds)
  - 8/10 rounds for ECHO-256/512

Overview

Related Work

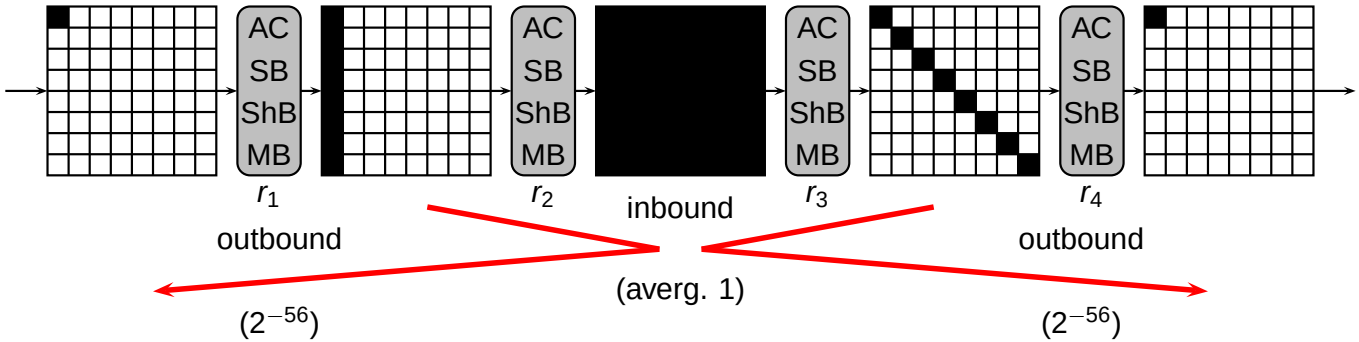
- Truncated Differences [Knudsen, FSE 1994]
  - on bytes: active or not active
- Attack on Grindahl Hash Function [Peyrin, Asiacrypt 2007]
  - truncated differences
  - probabilistic MixColumns propagation
- Rebound Attack [Mendel *et al.*, FSE 2009]
  - truncated differences
  - probabilistic MixColumns propagation
  - match-in-the-middle using S-box

Summary

- Overview of Results

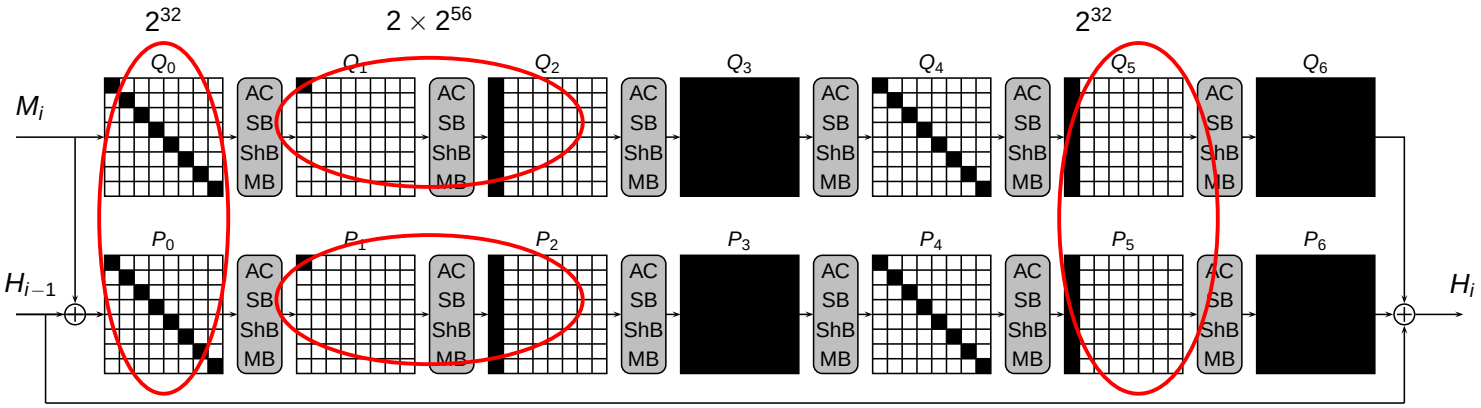
target	rounds	computational complexity	memory requirements	type	section
Grøstl-256	6	2^{112}	2^{64}	semi-free-start collision	[Mendel *et al.*]
	6	2^{64}	2^{64}	semi-free-start collision	this work
	7	2^{55}	2^{64}	permutation distinguisher	this work
ECHO	7	2^{896}	-	permutation distinguisher	ECHO Specification
	7	2^{384}	2^{64}	permutation distinguisher	this work
AES	7	2^{56}	-	known-key-distinguisher	[Knudsen]
	7	2^{24}	2^{16}	known-key-distinguisher	this work

Rebound Attack on Grøstl-256



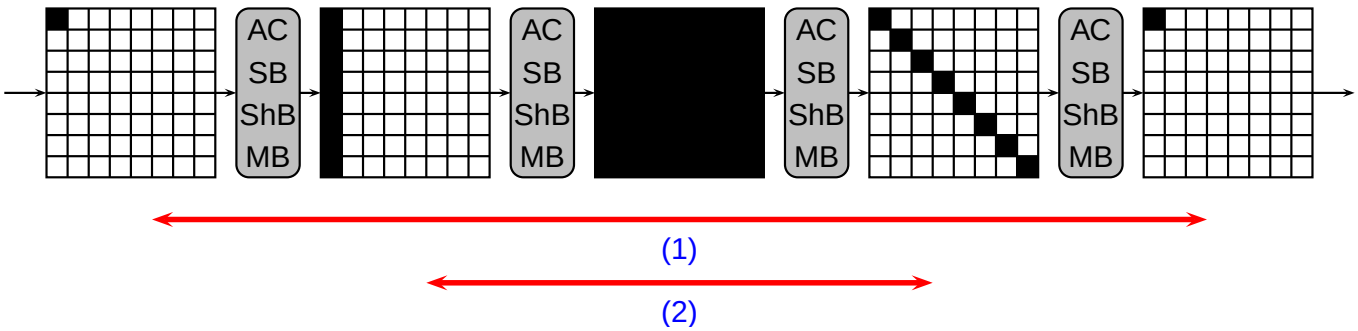
- Inbound phase:
  - (1) start with differences in round r_2 and r_3
  - (2) match-in-the-middle at S-box using values of the state
- Outbound phase:
  - (3) probabilistic propagation in MixBytes of r_1 and r_4
  - (4) match difference at input and output

Previous Results on Grøstl-256



- Semi-free-start collision on 6 rounds of Grøstl-256
  - only one 8 $\rightarrow$ 1 MixBytes transition
  - birthday match on 16-byte input/output differences
- Complexity of attack: $\sim 2^{120}$

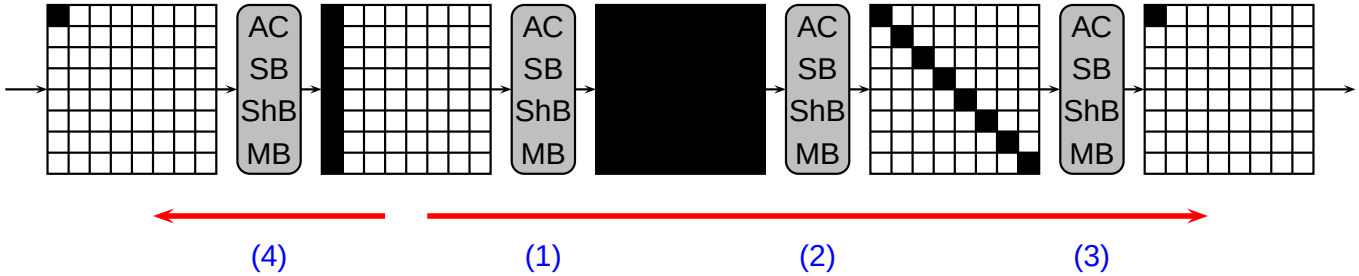
A Linearized Match-in-the-Middle Attack



(1) Filtering for differential paths
- using probabilities of SubBytes/ShiftBytes/MixBytes
- by guess-and-determine with complexity 1

(2) Solving for conforming state pairs
- S-box behaves linearly for fixed input/output differential
- solve linearly with complexity 2^{48} (previously: 2^{112})

A Start-from-the-Middle Technique



(1) Forward compute MixBytes independently
- but start with "good" differentials (backward)

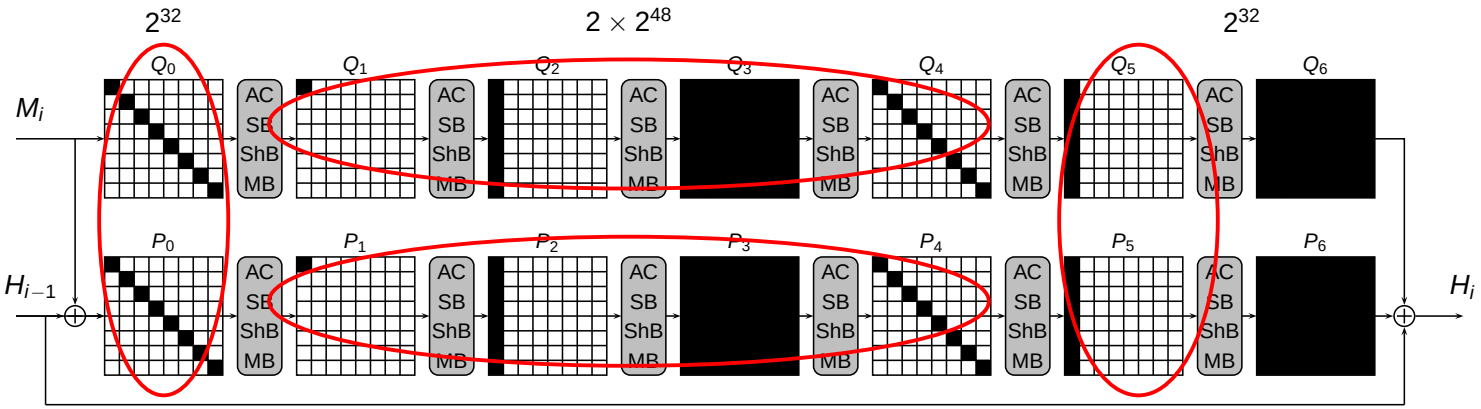
(2) Choose output of S-box to ensure 64 $\longrightarrow$ 8
- for each byte, we can choose from 127 differences

(3) Use remaining freedom in bytes to ensure 8 $\longrightarrow$ 1
- for each difference, use values (a, b) and (b, a)

(4) Probabilistic in MixBytes
- (total) complexity 2^{48}

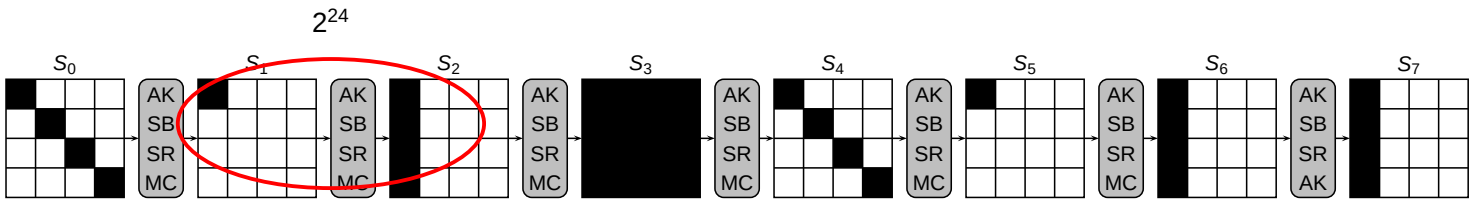
Overview

Results for Grøstl-256 Compression Function



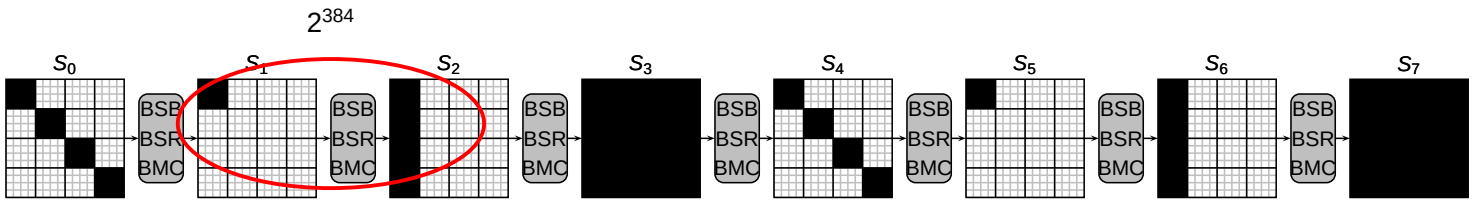
- Find pair following 4-round truncated path
  - complexity 2^{48}
- Semi-free-start collision on 6/10 rounds
  - complexity 2^{64}, memory 2^{64}
- Here: not enough freedom for 7-round collision attack
  - distinguisher on 7/10 rounds (complexity 2^{55})

Results for Known-Key AES



- Using Start-from-the-Middle Technique
- Find pair following 4-round truncated path
  - complexity 2^{24} (MixColumns)
  - memory 2^{16} (S-Box lookup table)
- Known-key distinguisher for 7-round AES

Results for ECHO Permutation



- Using Start-from-the-Middle Technique
- Find pair following 4-round truncated path
  - complexity 2^{384} (BigMixColumns)
  - memory 2^{64} (SuperBox lookup table)
- Improved distinguisher for 7-round permutation

Conclusions

- The Rebound Attack
  - simple and quick analysis
- This Improved Cryptanalyis
  - two different fine-tuned techniques
  - utilize almost all available degrees of freedom
- Future work
  - squeeze the last out of it
  - proof bounds (degrees of freedom vs. costs for SB/MC)
  - apply to other SHA-3 candidates

Conclusions

- The Rebound Attack
  - simple and quick analysis
- This Improved Cryptanalyis
  - two different fine-tuned techniques
  - utilize almost all available degrees of freedom
- Future work
  - squeeze the last out of it
  - proof bounds (degrees of freedom vs. costs for SB/MC)
  - apply to other SHA-3 candidates

Thank you for your Attention!